

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representation of
The original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

E5934

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-161514

(43)Date of publication of application : 18.06.1999

(51)Int.Cl.

G06F 11/28

(21)Application number : 09-330438

(71)Applicant : TOSHIBA CORP

(22)Date of filing : 01.12.1997

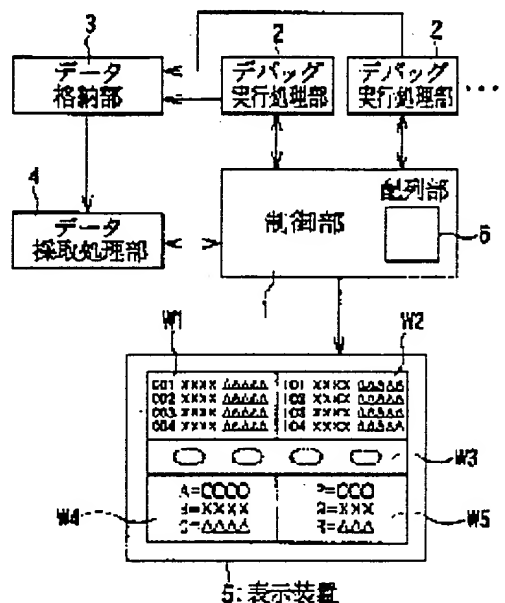
(72)Inventor : ONODERA HIKOJI

(54) DEBUGGER DEVICE, DEBUGGING METHOD AND STORAGE MEDIUM RECORDING DEBUGGING PROGRAM

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a debugger device capable of debugging respective modules without omission even in the case of simultaneously activating the plural modules.

SOLUTION: This debugger device is provided with a control part 1, plural debugging execution processing parts 2, a data storage part 3, a data sampling processing part 4 and a display device 5. In this case, when the module during debugging at present instructs the activation of a slave process, the control part 1 selects the debugging execution processing part 2 for debugging the module corresponding to the slave process and sets information required for debugging to the selected debugging execution processing part 2. Also, when the transmission and reception of data are instructed, the control part 1 instructs the module for transmitting and receiving the data to transfer the information relating to transmission/reception data to the data storage part 3 and then, instructs the data sampling processing part 4 to generate the data for screen display. Receiving the instruction, the data sampling processing part 4 generates the data for the screen display.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2000 Japanese Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-161514

(43) 公開日 平成11年(1999) 6月18日

(51) Int.Cl.⁶

識別記号

F I

G 0 6 F 11/28

G 0 6 F 11/28

P

審査請求 未請求 請求項の数 7 O L (全 6 頁)

(21) 出願番号 特願平9-330438

(22) 出願日 平成9年(1997)12月1日

(71) 出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72) 発明者 小野寺 彦 事

東京都府中市東芝町1番地 株式会社東芝
府中工場内

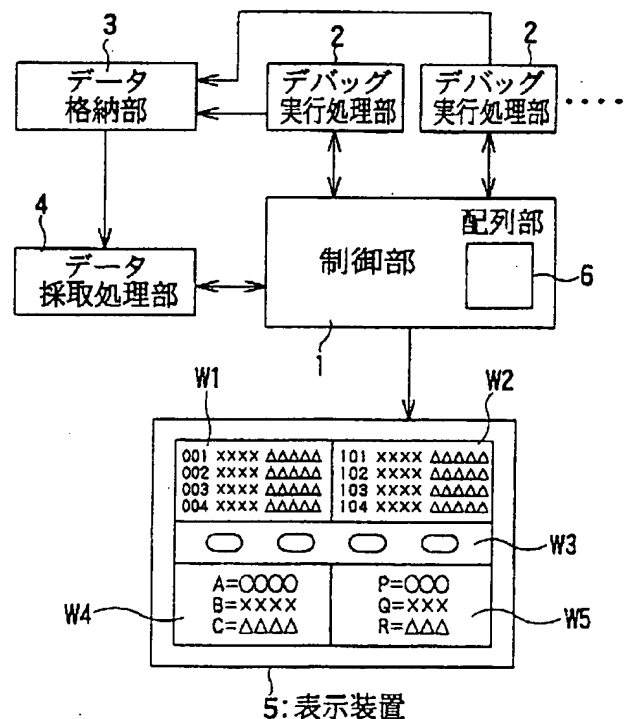
(74) 代理人 弁理士 佐藤 一雄 (外3名)

(54) 【発明の名称】 デバッグ装置、デバッグ方法、およびデバッグプログラムを記録した記録媒体

(57) 【要約】

【課題】 複数のモジュールを同時に起動する場合でも、各モジュールのデバッグを取りこぼしなく行うことができるデバッグ装置を提供する。

【解決手段】 本発明のデバッグ装置は、制御部1と、複数のデバッグ実行処理部2と、データ格納部3と、データ採取処理部4と、表示装置5とを備える。現在デバッグ中のモジュールが子プロセスの起動を指示すると、制御部1は、子プロセスに対応するモジュールのデバッグを行うデバッグ実行処理部2を選択し、選択したデバッグ実行処理部2に、デバッグに必要な情報を設定する。また、データの送受信が指示されると、制御部1は、データの送受信を行うモジュールに対して、送受信データに関する情報をデータ格納部3に転送するよう指示した後、データ採取処理部4に対して、画面表示用のデータの生成を指示する。この指示を受けて、データ採取処理部4は、画面表示用のデータを生成する。



【特許請求の範囲】

【請求項1】複数のモジュールからなるプログラムのデバッグを行うことが可能なデバッグ装置であって、モジュール単位でデバッグを行う複数のデバッグ実行処理手段と、

一つのモジュールから他のモジュールを起動する場合に、起動元のモジュールをデバッグする前記デバッグ実行処理手段とは別のデバッグ実行処理手段を起動先のモジュールのデバッグ用として割り当て、かつ、新たに割り当てた前記デバッグ実行処理手段に対してデバッグに必要な情報を設定するデバッグ制御手段と、を備えることを特徴とするデバッグ装置。

【請求項2】デバッグ対象であるモジュールの種類と、そのモジュールのデバッグを行う前記デバッグ実行処理手段の種類とを対応づけて記憶するデバッグ情報格納手段を備えることを特徴とする請求項1に記載のデバッグ装置。

【請求項3】複数のモジュールからなるプログラムのデバッグを行うことが可能なデバッグ装置であって、モジュール単位でデバッグを行う複数のデバッグ実行処理手段と、

前記複数のデバッグ実行処理手段のそれぞれに対してデバッグ対象となるモジュールを割り当て、かつ、モジュールを割り当てた前記デバッグ実行処理手段に対してデバッグに必要な情報を設定するデバッグ制御手段と、複数のモジュール間でデータの送受信を行う場合に、送信側モジュールの名称およびプログラム実行位置を含む送信側情報と、受信側モジュールの名称およびプログラム実行位置を含む受信側情報とを表示装置に表示させる表示制御手段と、を備えることを特徴とするデバッグ装置。

【請求項4】複数のモジュール間でデータの送受信を行う場合に、送受信の区別、送受信データ、送受信の対応関係を示す送受信キー、および送受信時のモジュール内でのプログラム実行位置を記憶する送受信情報記憶手段を備え、

前記表示制御手段は、前記送受信情報記憶手段に記憶されているデータのうち、前記送受信キーが一致するデータに基づいて、前記送信側情報と前記受信側情報とを前記表示装置の同一画面上に表示させることを特徴とする請求項3に記載のデバッグ装置。

【請求項5】前記複数のデバッグ実行処理手段は、高級言語で記述されたソースコードのデバッグを行うことを特徴とする請求項1～4のいずれかに記載のデバッグ装置。

【請求項6】複数のモジュールからなるプログラムのデバッグを行うことが可能なデバッグ方法であって、モジュール単位でデバッグを行う複数のデバッグ実行処理手段が設けられ、

一つのモジュールから他のモジュールを起動する場合に

は、起動元のモジュールをデバッグする前記デバッグ実行処理手段とは別のデバッグ実行処理手段を起動先のモジュールのデバッグ用として割り当て、かつ、新たに割り当てた前記デバッグ実行処理手段に対してデバッグに必要な情報を設定し、

複数のモジュール間でデータの送受信を行う場合には、送信側モジュールの名称およびプログラム実行位置を含む送信側情報と、受信側モジュールの名称およびプログラム実行位置を含む受信側情報とを表示装置の同一画面上に表示させることを特徴とするデバッグ方法。

【請求項7】複数のモジュールからなるプログラムのデバッグを行うことが可能なデバッグプログラムを記録したコンピュータ読み取り可能な記録媒体であって、モジュール単位でデバッグを行う複数のデバッグ実行処理手段と、

一つのモジュールから他のモジュールを起動する場合に、起動元のモジュールをデバッグする前記デバッグ実行処理手段とは別のデバッグ実行処理手段を起動先のモジュールのデバッグ用として割り当て、かつ、新たに割り当てた前記デバッグ実行処理手段に対してデバッグに必要な情報を設定するデバッグ制御手段と、

複数のモジュール間でデータの送受信を行う場合に、送信側モジュールの名称およびプログラム実行位置を含む送信側情報と、受信側モジュールの名称およびプログラム実行位置を含む受信側情報とを表示装置の同一画面上に表示させる表示制御手段と、を実行させるためのデバッグプログラムを記録したコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、複数のモジュールからなるプログラムのデバッグを行う技術に関し、特に、C言語等の高級言語で記述されたプログラムのデバッグを対象とする。

【0002】

【従来の技術】C言語などの高級言語で記述されたプログラムに誤りがあるか否かを、ソースコードレベルで検証するデバッグ装置が広く利用されている。この種のデバッグ装置は、プログラムを構成するソースコードを各行ごとに実行して、その実行結果を画面に表示する機能を有する。この種のデバッグ装置を利用すれば、ソースコードを行単位で検証できるため、バグを容易に見え、プログラムの開発期間を短縮することができる。

【0003】

【発明が解決しようとする課題】最近では、CPUの高速化に伴って、プログラム（ソフトウェア）も複雑高度化する傾向にあり、プログラムの開発期間も長くなる傾向にある。プログラムを効率的に開発するには、プログラムを機能単位で複数のモジュールに分け、各モジュールごとに開発を行うのが望ましい。

【0004】ところが、従来のデバッグ装置は、複数のモジュールを同時にデバッグすることができず、例えば、あるモジュールから子プロセスを起動する場合には、その子プロセスのデバッグを行うデバッグ装置を新たに起動する必要があった。また、異なる複数のモジュール間でデータの送受信を行う場合も、送信側モジュールと受信側モジュールで別個にデバッグ装置を起動しなければならなかった。

【0005】このため、場合によっては、デバッグ装置を起動している最中に、子プロセスの処理やデータ送受信が終了してしまうことがあり、子プロセスの処理やデータ送受信が正常に行われたか否かを確認できないおそれがあった。

【0006】本発明は、このような点に鑑みてなされたものであり、その目的は、子プロセスの起動やデータの送受信など、複数のモジュールを同時に起動した場合でも、各モジュールのデバッグを取りこぼしなく行うことができるデバッグ装置を提供することにある。

【0007】

【課題を解決するための手段】上述した課題を解決するために、請求項1の発明は、複数のモジュールからなるプログラムのデバッグを行うことが可能なデバッグ装置であって、モジュール単位でデバッグを行う複数のデバッグ実行処理手段と、一つのモジュールから他のモジュールを起動する場合に、起動元のモジュールをデバッグする前記デバッグ実行処理手段とは別のデバッグ実行処理手段を起動先のモジュールのデバッグ用として割り当て、かつ、新たに割り当てた前記デバッグ実行処理手段に対してデバッグに必要な情報を設定するデバッグ制御手段と、を備えるものである。

【0008】請求項1の発明を、例えば図1に対応づけて説明すると、「複数のデバッグ実行処理手段」はデバッグ実行処理部2に、「デバッグ制御手段」は制御部1に、それぞれ対応する。

【0009】請求項2の発明を、例えば図1に対応づけて説明すると、「デバッグ情報格納手段」はデータ格納部3に対応する。

【0010】請求項3の発明を、例えば図1に対応づけて説明すると、「表示制御手段」は制御部1およびデータ採取処理部4に対応する。

【0011】請求項4の発明を、例えば図1に対応づけて説明すると、「送受信情報記憶手段」はデータ格納部3に対応する。

【0012】

【発明の実施の形態】以下、本発明を適用したデバッグ装置について、図面を参照しながら具体的に説明する。以下に説明するデバッグ装置は、高級言語で記述された複数のモジュールからなるプログラムのソースコードデバッグを行うものである。

【0013】図1は本発明に係るデバッグ装置の一実施

形態のブロック図である。図1のデバッグ装置は、制御部1と、複数のデバッグ実行処理部2と、データ格納部3と、データ採取処理部4と、表示装置5とを備える。

【0014】デバッグ実行処理部2はいずれも同じように構成され、制御部1から指示されたプログラムモジュールのソースコードデバッグを行う。デバッグ結果は制御部1を介して表示装置5に表示される。制御部1の内部には、図示のように配列部6が設けられ、この配列部6には、新たに起動されたモジュールの名称（より詳細には、モジュールのプロセスID）と、そのモジュールのデバッグを行うデバッグ実行処理部2の名称とが対応づけて格納される。なお、配列部6は、具体的には、メモリやレジスタなどで構成される。

【0015】データ格納部3は、複数のモジュール間で送受信されるデータ等を格納する。データ採取処理部4は、データ送受信時の各モジュールのプログラム実行位置等を表示装置5に表示するためのデータを生成する。データ採取処理部4で生成されたデータは、制御部1を介して表示装置5に表示される。

【0016】図2は図1の制御部1の処理動作を示すフローチャートであり、以下、このフローチャートに基づいて、図1の制御部1の処理動作を説明する。まず、制御部1は、プログラムのメインモジュールのデバッグを行うデバッグ実行処理部2を選択し、選択したデバッグ実行処理部2に、デバッグに必要な情報を設定する（ステップS1）。このデバッグ実行処理部2は、メインモジュールのソースコードを1行ずつ実行して、行単位でデバッグを行う。

【0017】次に、制御部1は、デバッグ実行処理部2が行ったメインモジュールのデバッグ結果を表示装置5に表示する制御を行う（ステップS2）。次に、制御部1は、現在デバッグ中のメインモジュールが子プロセスの起動を指示したか否かを判定する（ステップS3）。子プロセスの起動が指示されると、制御部1は、子プロセスに対応するモジュールのデバッグを行うデバッグ実行処理部2を選択し、選択したデバッグ実行処理部2に、デバッグに必要な情報を設定する（ステップS4）。同時に、制御部1は、新たに起動した子プロセスのプロセスIDと、そのモジュールのデバッグを行うデバッグ実行処理部2の名称とを、配列部6に格納する。この配列部6には、新たに子プロセスを起動するたびに、その子プロセスのプロセスIDと、デバッグ実行処理部2の名称とが追加して格納される。

【0018】次に、制御部1は、ステップS4で選択したデバッグ実行処理部2が行ったデバッグ結果を表示装置5に表示する（ステップS5）。

【0019】図3はメインモジュールから子プロセスを起動する場合のデバッグ装置内でのデータの流れを説明する図である。制御部1は、まず、メインモジュールのプロセスID「a a」と、メインモジュールのデバッグ

を行うデバッグ実行処理部2の名称「D1」とを対応づけて配列部6に格納する。また、デバッグ実行処理部2は、メインモジュールaaを1行ずつ実行してデバッグを行い、デバッグ結果を制御部1に転送する。

【0020】新たなモジュールを起動する場合には、制御部1は、そのモジュールのプロセスID「bb」と、そのモジュールのデバッグを行うデバッグ実行処理部2の名称「D2」とを対応づけて配列部6に追加して格納する。一方、デバッグ実行処理部2は、モジュールbbを1行ずつ実行してデバッグを行い、デバッグ結果を制御部1に転送する。

【0021】また、モジュールbbが子プロセスを起動した場合には、制御部1は、子プロセスに対応するモジュールのプロセスID「cc」と、そのモジュールのデバッグを行うデバッグ実行処理部2の名称「D3」とを対応づけて配列部6に追加して格納する。

【0022】このように、制御部1は、新たなプロセスや子プロセスが起動されるたびに、そのプロセスのプロセスIDと、そのプロセスに対応するモジュールのデバッグを行うデバッグ実行処理部2の名称とを、配列部6に順次追加して格納する。

【0023】また、制御部1は、各デバッグ実行処理部2から転送されたデバッグ結果を表示装置5に表示する。図3に示すように、表示装置5の表示領域W1には、メインモジュールのソースコードが、表示領域W2には、メインモジュールから起動された子プロセスに対応するモジュールのソースコードが表示される。また、表示領域W3には、画面に表示する情報の選択ボタンや、デバッグの開始・停止指示ボタンなどが表示される。また、表示領域W4には、メインモジュールの現在の実行位置における変数値などが表示され、表示領域W5には、子プロセスに対応するモジュールの現在の実行位置における変数値などが表示される。

【0024】図3では、2つのモジュールのデバッグ結果を表示しているが、3つ以上のモジュールのデバッグ結果を同時に表示してもよい。ただし、一画面に表示可能な情報には限りがあるため、操作者の指示により、表示する情報を任意に選択できるようにするのが望ましい。例えば、3つ以上のモジュールが同時に起動している場合には、デバッグ結果を表示させたいモジュールに対応するデバッグ実行処理部2を配列部6の中から検索して表示装置5に表示する。

【0025】図2に戻って、ステップS5の処理が終了した場合、または、現在デバッグ中のモジュールが子プロセスの起動を指示しなかった場合には、制御部1は、現在デバッグ中のモジュールが他のモジュールに対してデータの送受信を指示したか否かを判定する(ステップS6)。データの送受信が指示された場合には、制御部1は、データの送受信を行うモジュールに対して、送受信データに関する情報をデータ格納部3に転送するよう

指示する(ステップS7)。

【0026】図4はデータ格納部3に格納される情報の概要を説明する図である。データ格納部3には、送受信の区別を示す情報I1と、送受信データI2と、送信データと受信データが一致するか否かを判別するのに用いられる送受信キーI3と、データの送受信を行うモジュールが属するプログラム名I4と、データの送受信を行うモジュールのソースコード名I5と、現在実行中のソースコード行I6とが格納される。データを送信するモジュールとデータを受信するモジュールは、双方とも、図4のような情報をデータ格納部3に格納する。

【0027】図2のステップS7の処理が終了すると、次に、制御部1は、データ採取処理部4に対して、画面表示用のデータの生成を指示する。この指示を受けて、データ採取処理部4は、データ格納部3から送受信キーが同じ情報を抽出して、画面表示用のデータを生成する(ステップS8)。

【0028】図5はデータ採取処理部4が生成する画面表示用のデータの一例を示す図である。図示のように、データ採取処理部4は、送信側モジュールが属するプログラム名と、送信側モジュールのソースコード名と、送信側モジュールが実行中のソースコード行と、受信側モジュールが属するプログラム名と、受信側モジュールのソースコード名と、受信側モジュールが実行中のソースコード行とを、1行にまとめたデータを生成する。図5のデータを生成した後は、データ格納部3に同じデータを格納しておく必要がないため、これらデータはデータ格納部3から削除される。

【0029】データ採取処理部4で画面表示用のデータが生成されると、次に、制御部1は、そのデータを表示装置5に表示する(ステップS9)。ステップS9の処理が終了した場合、あるいは、ステップS6で、データの送受信が指示されなかった場合には、制御部1はステップS3以降の処理を繰り返す。

【0030】図6は異なるモジュール間でデータの送受信を行う場合のデバッグ装置のブロック図である。図6では、モジュールaaからモジュールbbに対してデータを送信する例を示している。制御部1は、モジュールaaのデバッグを行うデバッグ実行処理部2からデータの送信が通知されると、各デバッグ実行処理部2に対して、送信データや受信データなどをデータ格納部3に転送するように指示する。この通知を受けて、各デバッグ実行処理部2は、送受信データや現在実行中のソースコード行などの情報をデータ格納部3に転送する。

【0031】また、データ採取処理部4は、制御部1からの指示に応じて、データ格納部3から、モジュール間のデータ送受信に関する情報を抽出して、図5に示すような表示用のデータを生成して、制御部1に転送する。制御部1は転送されたデータを表示装置5に表示する。

【0032】このように、本実施形態は、デバッグ実行

処理部2を複数備えるため、一つのモジュールが子プロセスを起動する場合でも、その子プロセスのデバッグを即座に行うことができ、複数のモジュールに対するデバッグ結果を取りこぼしなく、同一画面上に表示することができる。また、各デバッグ実行処理部2は、入力条件を変えることで、どのようなモジュールのデバッグも行うことができるため、複数のデバッグ装置を設ける必要がなくなる。

【0033】また、異なるモジュール間でデータの送受信を行う場合には、送受信したデータや、送信側モジュールや受信側モジュールのソースコード行などを同時に表示するようにしたため、データ送受信時のデバッグを詳細かつ簡易に行うことができる。

【0034】図1に示したデバッグ装置内の制御部1、デバッグ実行処理部2、データ格納部3、およびデータ採取処理部4は、それぞれハードウェアで実現してもよく、あるいはソフトウェアで実現してもよい。また、本発明をソフトウェアで実現した場合には、そのソフトウェアをフロッピディスクやCD-ROM等の各種の記録媒体に記録して、コンピュータで実行させるようにしてもよい。

【0035】また、デバッグ結果の表示形式は図1等に示したものに限定されない。また、デバッグ実行処理部2の数も2個以上であれば、特に制限はない。さらに、図1等では、制御部1が表示装置5の表示制御を行う例を説明したが、データ採取処理部4に表示制御を行わせてもよい。

【0036】

【発明の効果】以上詳細に説明したように、本発明によれば、モジュール単位でデバッグを行う複数のデバッグ実行処理手段を設け、子プロセスが起動されると、その

子プロセスに対応するモジュールに対して即座にデバッグ実行処理手段を割り当てるようにしたため、子プロセスの起動と同時に、その子プロセスのデバッグを行うことができ、全ソースコードを取りこぼしなく、デバッグすることができる。

【0037】また、異なるモジュール間でデータの送受信を行う場合には、送信側モジュールの送信側情報と、受信側モジュールの受信側情報とを同一画面上に表示するようにしたため、データ送受信時のデバッグを詳細かつ簡易に行うことができる。

【図面の簡単な説明】

【図1】本発明に係るデバッグ装置の一実施形態のブロック図。

【図2】図1の制御部の処理動作を示すフローチャート。

【図3】メインモジュールから子プロセスを起動する場合のデバッグ装置内でのデータの流れを説明する図。

【図4】データ格納部に格納される情報の概要を説明する図。

【図5】データ採取処理部が生成する画面表示用のデータの一例を示す図。

【図6】異なるモジュール間でデータの送受信を行う場合のデバッグ装置のブロック図。

【符号の説明】

- 1 制御部
- 2 デバッグ実行処理部
- 3 データ格納部
- 4 データ採取処理部
- 5 表示装置
- 6 配列部

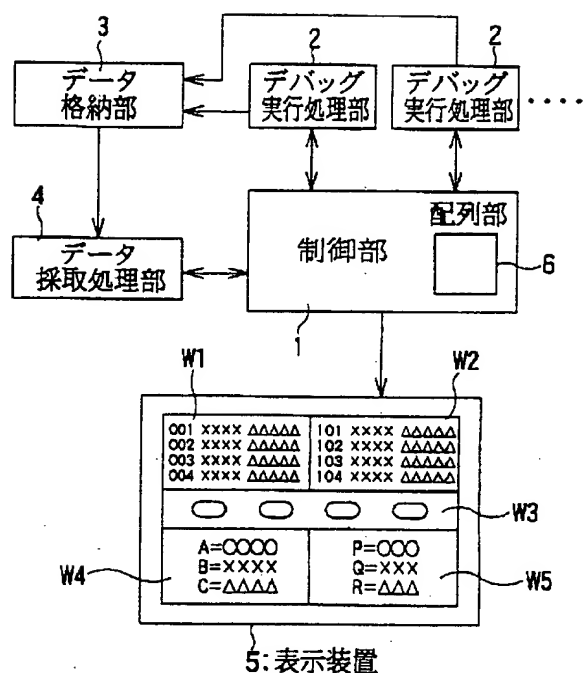
【図4】

送受信 区別	送信・受信データ	送受信 キー	プログラム 名	該当 ソースコード名	ソースコード 行
I1	I2	I3	I4	I5	I6

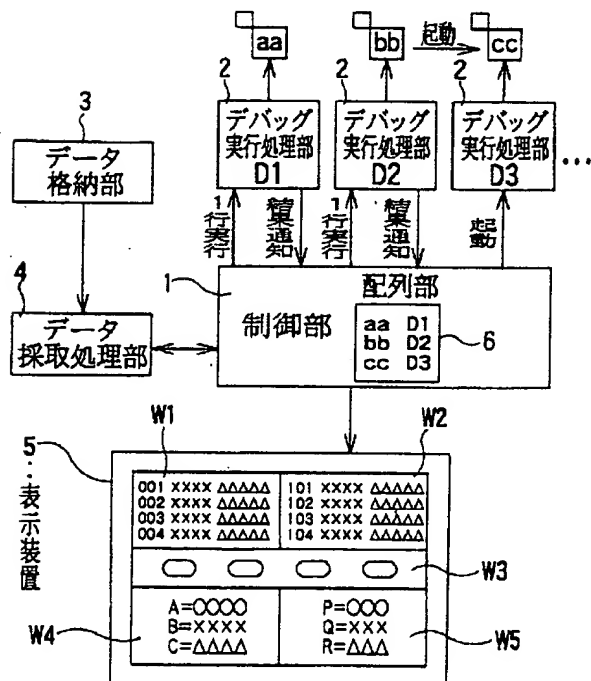
【図5】

送信側			受信側		
プログラム 名	ソース コード名	ソース コード行	プログラム 名	ソース コード名	ソース コード行
abcd	abcd.c	145	efghi	efghi.c	201

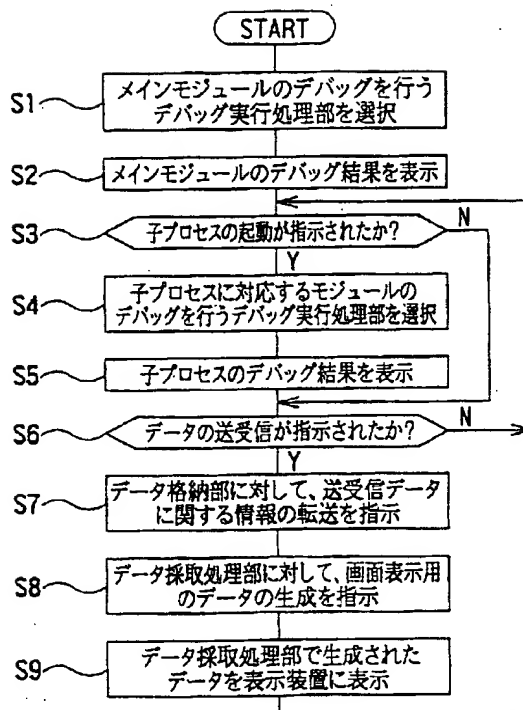
【図1】



【図3】



【図2】



【図6】

